Microsoft CryptoAPI Overview and an (fine) Example of CAPI Application Development

Aaron Margosis

Microsoft Consultant

Patrick Arnold

Federal Program Manager

Security, Standards, Strategic Programs

Microsoft Public Key Security Objectives

- Create a flexible, comprehensive, and efficient security infrastructure
  - Enterprise, Intranet, and Internet
- Support both public and secret key technology
- Provide integrated Enterprise administration
- Ensure interoperability (through adherence with standards)

CryptoAPI (1 of 5)

- Foundation for PK Security
  - Part of Internet Explorer since 3.x and part of Windows NT 4.0 since SP2
- Cryptography
  - Service provider model
  - Flexible algorithm & key length support
- Certificate Management
  - Mgmt and storage for x509v3 Certs
- Industry Standard Messaging

CryptoAPI (2 of 5)

- Cryptographic services
  - Key Generation and Management
  - Hashing
  - Digital Signatures and Verification
  - Key Exchange
  - Bulk Encryption/ Decryption
- CSPs configure local environment
  - consistent APIs
  - isolate applications from export/import requirements

CryptoAPI (3 of 5)

- CSPs
  - Implement specific algorithms and key strength
  - Software or hardware based
- Microsoft software providers
  - Base provider (512-bit RSA, 40-bit RC2/RC4)
  - Enhanced provider (>512-bit RSA, 128-bit RC2/RC4, DES, 3DES) for N.A. only
  - DSS and D-H provider

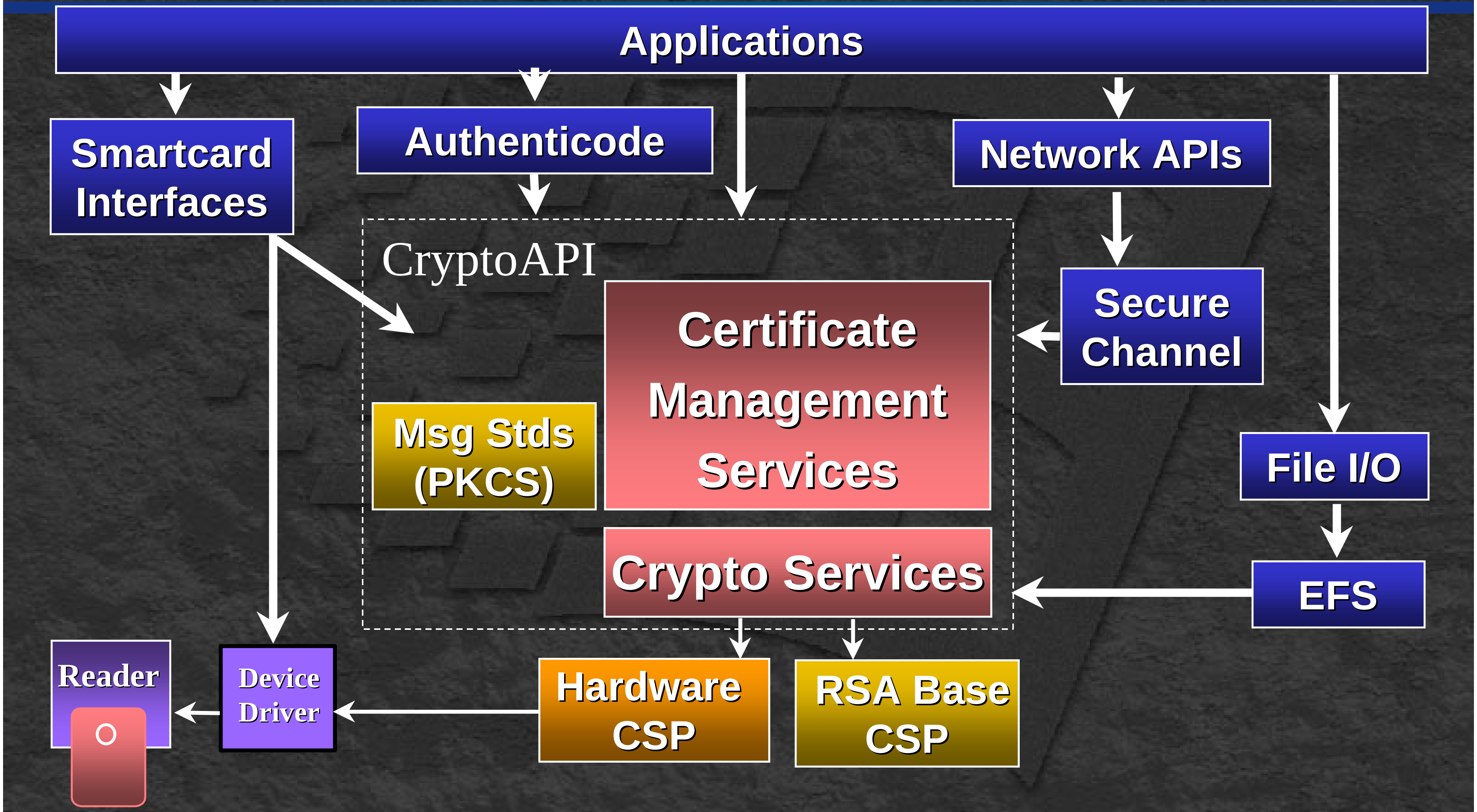
CryptoAPI (4 of 5)

- Certificate handling
  - x509v3 Certs & Cert chains
  - Certificate parsing (ASN.1, common extensions)
- Certificate Storage
  - Protected repository for CA root Certs and storage of intermediate issuing CAs
  - User store for personal Certs
  - Manages binding between Cert & Key set
  - Roaming and backup support via the Windows 2000 Active Directory or other 3rd LDAP directory

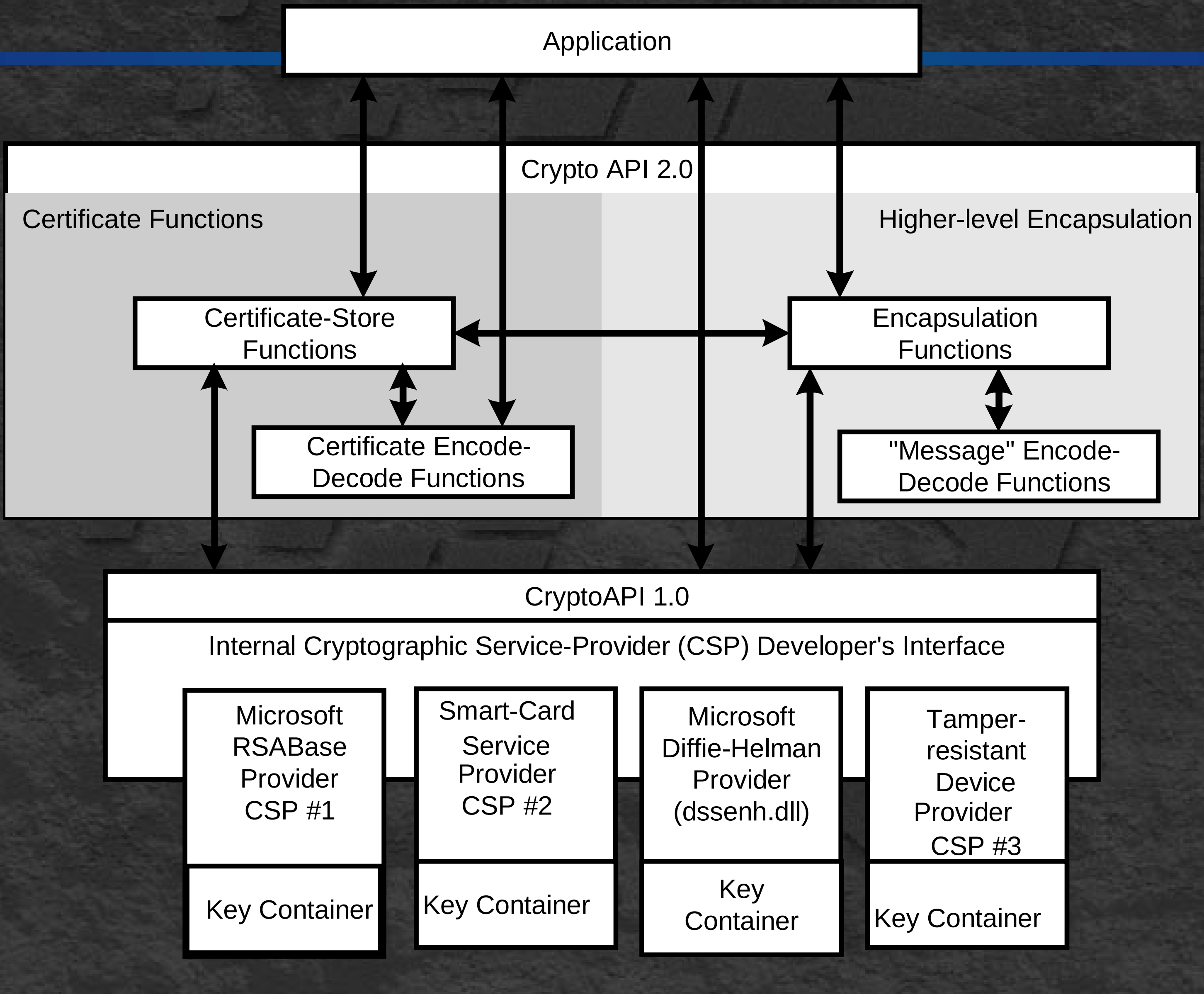
CryptoAPI (5 of 5)

- **Std Messages for Cert Enrollment**
  - PKCS 10 - Certificate request
  - PKCS 7
    - Certificates and certificate chains
    - Collections of authenticated attributes
- **PKCS 9 - Countersignatures**
  - ex: Authenticode timestamps
- **PKCS #12 - Certificate & Key backup**

PK Components



CryptoAPI 2.0 Architecture


Application
Crypto API 2.0
Certificate Functions
Higher-level Encapsulation
Certificate-Store Functions
Encapsulation Functions
Certificate Encode-Decode Functions
"Message" Encode-Decode Functions
CryptoAPI 1.0
Internal Cryptographic Service-Provider (CSP) Developer's Interface
Microsoft RSABase Provider CSP #1
Key Container
Smart-Card Service Provider CSP #2
Key Container
Microsoft Diffie-Helman Provider (dssenh.dll)
Key Container
Tamper-resistant Device Provider CSP #3
Key Container

Support for Microsoft CAPI

- Atalla (a Tandem Company)
- BBN Corporation
- Cylink
- Datakey
- E-Lock Technologies
- Flat Connections
- Hewlett Packard
- Information Resource Engineering (IRE)
- Microsoft
- PC/SC Workgroup
- Querisoft
- Rainbow Technologies
- Real Software
- SPYRUS
- Trusted Information Systems, Inc. (TIS)

“DocSigner”

Digital Signing of Structured Storage Documents

Aaron J Margosis

Microsoft Consulting Services,

Federal Practice

What Is “DocSigner”?

- Digitally signs Word, Excel or other structured storage documents
- Embeds signature and certificate in the document
- Guarantees document authenticity, no matter where it goes

Structured Storage

- Formerly called “OLE Structured Storage”, “OLE compound files”
- “File system within a file”
- “Storages” and “streams” analogous to directories and files

Signing Algorithm (1)

- Choose a certificate
- Create a hash (`CryptCreateHash`)
- Recursive, starting from root storage
- List storage elements, sorted by name
- Hash each element’s properties structure (`CryptHashData`)
- If stream, hash contents;
  If storage, recurse

Signing Algorithm (2)

- Sign (encrypt) the hash (`CryptSignHash`)
- Create a special storage in the root, and add two streams
- Serialize the cert into one stream, the signed hash into the other

Verification Algorithm

- Deserialize certificate and signed hash from special storage
- Recalculate hash
- Using the cert’s public key, compare the signed hash to the new hash (`CryptVerifySignature`)

DocSigner as COM DLL

Can be invoked from:

- Standalone executable
- Within Microsoft Office (using VBA)
- Web page (script)
- WSH script
- Any other COM-enabled tool or environment